

1.Introduction

1.1: Overview

Now a day's Information and communication technology (ICT) plays a great role in different field or areas, thus Health care system belongs to this. This leads to various studies and researches being conducted to selected health care facilities. It is necessary to ensure a technology appropriate, equitable, affordable, efficient, and environmentally adaptable and consumer friendly system, designed to fully utilize the ICT for the maximum benefit in the health care industry.

Here computers have great relevant on storing data's securely and ease access on them in a short period of time. In order to exploit the ICT in health care system, Medical Information Assistance Management System is being built. It is robust, integrated technology. Medical Information Assistance Management System (M.I.A) is to provide a single platform of interaction for three major sections of medical industry which are Doctors, their Patients and Pharmacists and to manage their database. This will enhance the efficiency of clinical work and ease patient's and pharmacist's convenience,

1.2: Objectives

- To provide a single platform of interaction for three major sections of medical industry
- To keep track of healthcare information of patients.
- To manage the doctors' database.
- To help pharmacists to manage their medicinal inventory.
- Provides security to the data of each individual by using password for login
- To provide a user-friendly Graphical User Interface (GUI).
- To develop an accurate and flexible system, it will eliminate data redundancy.

1.2 Limitations of Existing System:

- Time consumption in data entry, as the records are to be manually maintained it consumes a lot of time.
- Lot of paper work is involved as the records are maintained in the files and registers.
- Storage Requires as files and registers are used the storage space requirement is increased.
- Less Reliable use of papers for storing valuable data information is not at all reliable.
- We cannot generate PDF for prescription and report

2. SYSTEM SPECIFICATION

2.1 Software Requirement Specification

- Requirement Gathering
 - i. Fact Finding Tools
- Feasibility Study
 - i. Technical
 - ii. Behavioural/ Operational
 - iii. Economical
- System Analysis
 - i. Diagrams
- System Design
 - i. Database Design
 - ii. Data Dictionary
 - iii. User Interface Design
 - iv. Forms (Input Screen)
 - v. Reports (Output Screen)

2.2 Software Requirements and Hardware Requirements

Software tool used :

- Front-End: HTML, CSS
- Operating System: Ubuntu 19.04, (works on windows as well)
- Back-End: Python
- Database: MySQL

Hardware Requirements :

- Minimum 2GB of RAM or more
- 2.5 GB of available hard-disk space
- Computer with a 1.1 GHz or faster processor

Project Design

3.1: Database Schema

1. Doctors

<u>Doc_id</u>	Name	Specialization	Experience	Clinic	Qualification	Contact	Patient_id	Passwd	rating
---------------	------	----------------	------------	--------	---------------	---------	------------	--------	--------

2. Patient

<u>Pat_id</u>	Name	Contact	Insurance_id	Medical_info	Doc_Id	passwd
---------------	------	---------	--------------	--------------	--------	--------

3. Appointment

<u>doctorID</u>	<u>date</u>	startTime	endTime	Pat_id
-----------------	-------------	-----------	---------	--------

4. Drugs

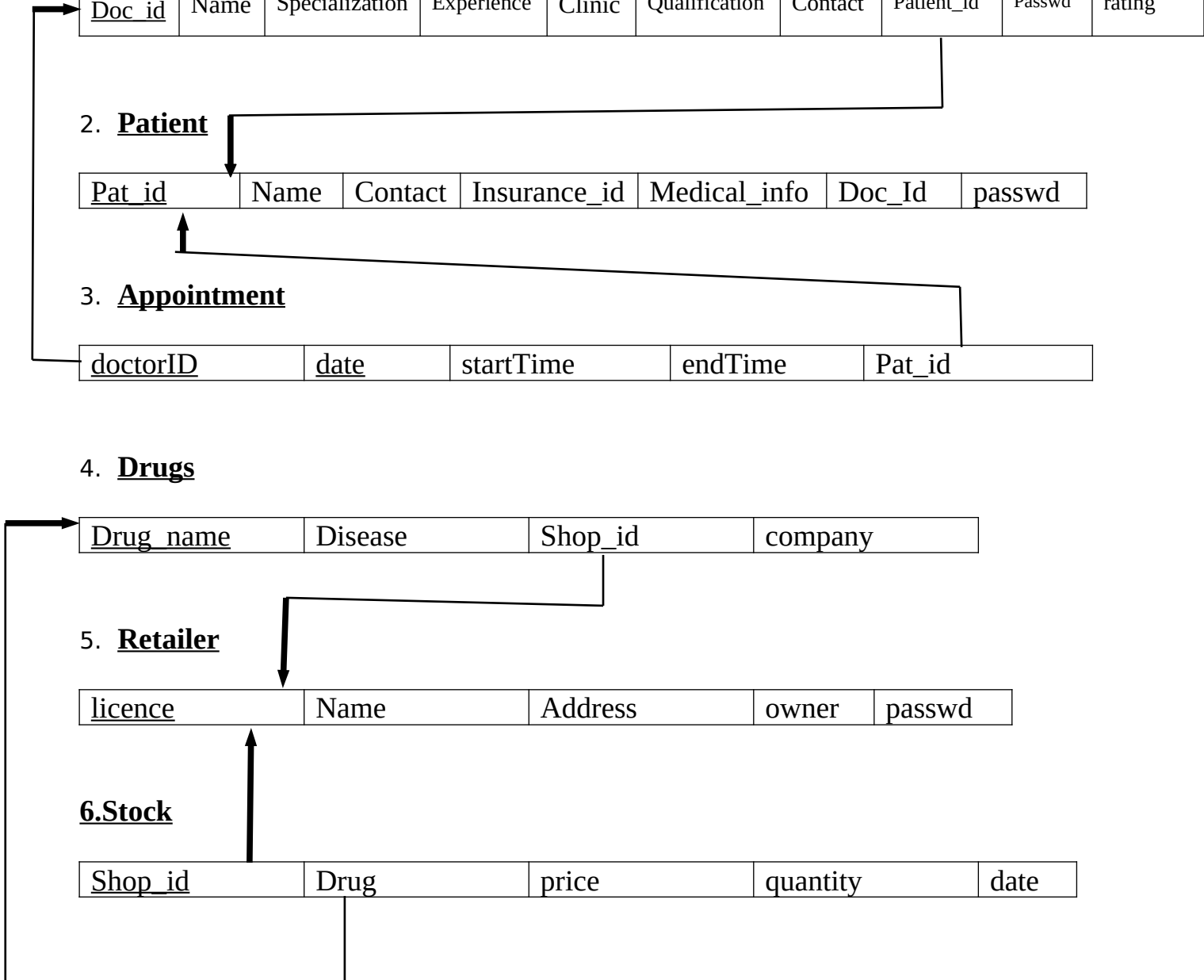
<u>Drug_name</u>	Disease	Shop_id	company
------------------	---------	---------	---------

5. Retailer

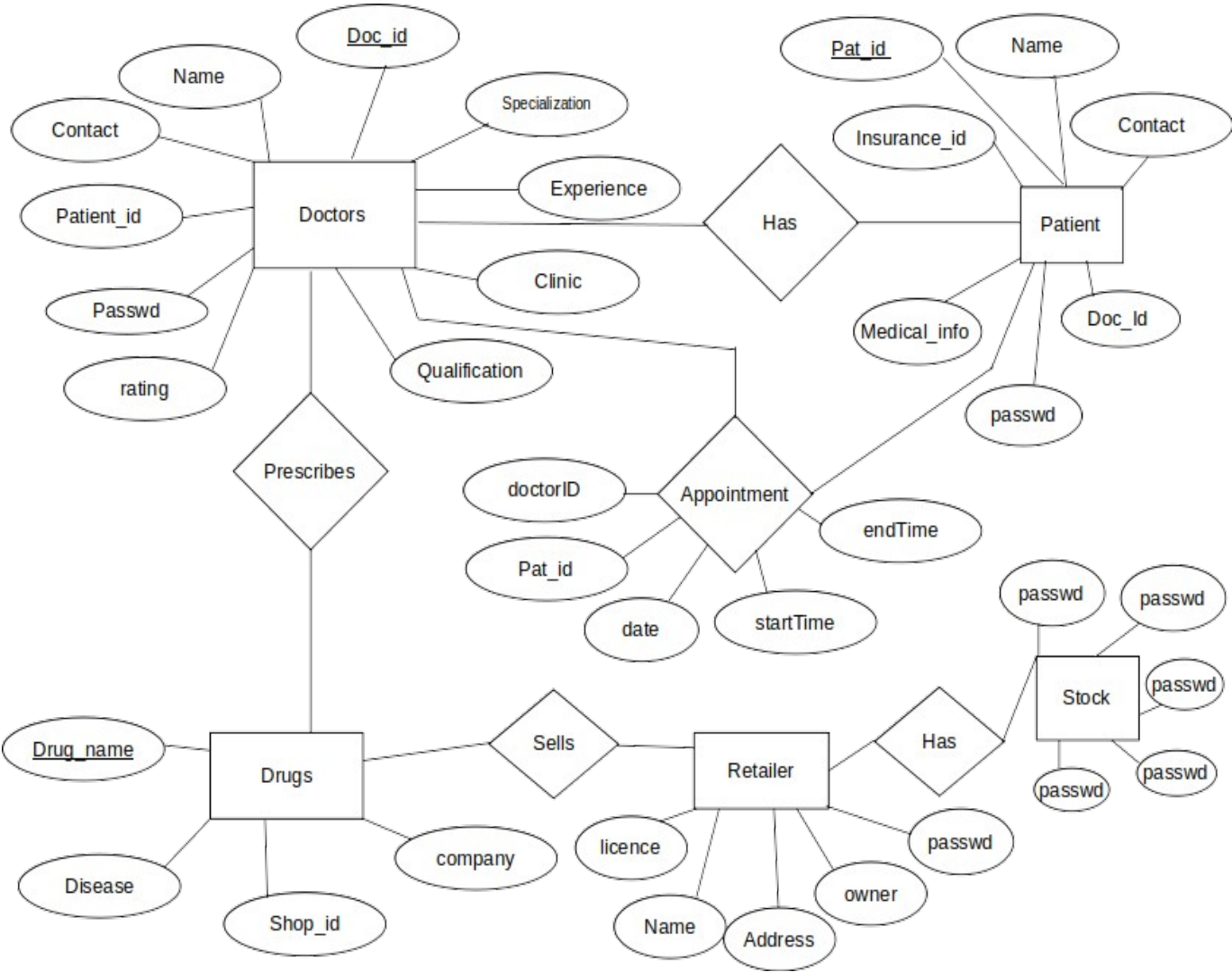
<u>licence</u>	Name	Address	owner	passwd
----------------	------	---------	-------	--------

6. Stock

<u>Shop_id</u>	Drug	price	quantity	date
----------------	------	-------	----------	------



3.2 E.R. Diagram



Implementation

4.1 Introduction

Python:

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

- Python can be treated in a procedural way, an object-orientated way or a functional way.

Database:

A Database Management System (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems.

Typical examples of DBMS use include accounting, human resources and customer support systems. Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database.

MYSQL is a open source relational database management system (RDBMS) based on Structured Query Language (SQL)

SQL:

Structured Query Language (SQL) is the language used to manipulate relational databases. SQL is tied very closely with the relational model.

In the relational model, data is stored in structures called relations or tables.

SQL statements are issued for the purpose of:

- Data definition: Defining tables and structures in the database (DDL used to create, alter and drop schema objects such as tables and indexes)
- Data Manipulation: Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.
- Transaction Control: Language which deals with a transaction within a database.

Database :

Tables Creation

```
CREATE TABLE `Doctors` (  
  `Doc_id` varchar(45) NOT NULL,  
  `Name` varchar(45) NOT NULL,  
  `Specialization` varchar(45) NOT NULL,  
  `Experience` int(11) NOT NULL,  
  `Clinic` varchar(45) NOT NULL,  
  `Qualification` varchar(45) NOT NULL,  
  `Contact` int(11) DEFAULT NULL, UNIQUE  
  `Patient_id` varchar(45) DEFAULT NULL,  
  `Passwd` varchar(45) NOT NULL,  
  `rating` int(11) DEFAULT NULL,  
  PRIMARY KEY (`Doc_id`)  
);
```

```
CREATE TABLE `Drugs` (  
  `Drug_name` varchar(45) NOT NULL, PRIMARY KEY  
  `Disease` varchar(45) NOT NULL,  
  `Shop_id` varchar(45) DEFAULT NULL,  
  `company` varchar(45) NOT NULL,  
  FOREIGN KEY (`Shop_id`) REFERENCES `Retailer` (`licence`) ON DELETE NO ACTION  
  ON UPDATE NO ACTION;  
);
```

```
CREATE TABLE `Patient` (  
  `Pat_id` varchar(45) NOT NULL, PRIMARY KEY  
  `Name` varchar(45) NOT NULL,  
  `Contact` int(11) DEFAULT NULL, UNIQUE  
  `Insurance_id` varchar(45) NOT NULL,, UNIQUE
```


`Medical_info` varchar(45) DEFAULT NULL,

`Doc_Id` varchar(45) DEFAULT NULL,

`passwd` varchar(45) NOT NULL,

FOREIGN KEY (`Doc_Id`) REFERENCES `Doctors` (`Doc_id`) ON DELETE CASCADE ON UPDATE CASCADE;

);

CREATE TABLE `Retailer` (

`licence` varchar(45) NOT NULL, PRIMARY KEY,

`Name` varchar(45) NOT NULL,

`Address` varchar(45) DEFAULT NULL,

`owner` varchar(45) NOT NULL,

`passwd` varchar(45) NOT NULL

);

CREATE TABLE `Stock` (

`Shop_id` varchar(45) DEFAULT NULL,

`Drug` varchar(45) NOT NULL,

`price` int(11) NOT NULL,

`quantity` int(10) DEFAULT NULL,

`date` varchar(45) DEFAULT NULL

FOREIGN KEY (`Drug`) REFERENCES `Drugs` (`Drug_name`) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (`Shop_id`) REFERENCES `Retailer` (`licence`) ON DELETE CASCADE ON UPDATE CASCADE;

);

CREATE TABLE `Appointment` (

`doctorID` varchar(45) NOT NULL,

`date` date NOT NULL,

```

startTime` time NOT NULL,
endTime` time NOT NULL,
`Pat_id` varchar(45) DEFAULT NULL
FOREIGN KEY (`doctorID`) REFERENCES `Doctors` (`Doc_id`) ON DELETE CASCADE ON
UPDATE CASCADE,
` FOREIGN KEY (`Pat_id`) REFERENCES `Patient` (`Pat_id`) ON DELETE CASCADE ON UPDATE
CASCADE
);

```

4.2: Stored Procedures :

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system such that it can be reused and shared by multiple programs.

The stored procedure used for this project is as follows:

ROUTINE NAME : noofpatients

TYPE : *PROCEDURE*

DIRECTION : *IN*

PARAMETER : *Doc_id*

DEFINITION:

DELIMITER \$\$

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `noofpatients` (IN `Doc_id` VARCHAR(45)
OUT nocount varchar(45)) BEGIN
select count(Pat_id) into nocount from Patient
where Doc_id = Doc_id;
END$$

```

4.3: Triggers :

A trigger is a special kind of stored procedure that automatically executes when an event occurs in database server.

The triggers used in our project are as follows:

1. TRIGGERNAME: ensureNewAppointmentsDoNotClash

TABLE: Appointment

TIME : BEFORE

EVENT: INSERT

DEFINATION:

DELIMITER \$\$

```
CREATE TRIGGER `ensureNewAppointmentsDoNotClash` BEFORE INSERT ON
`Appointment` FOR EACH ROW BEGIN
```

```
IF NOT slotIsAvailable(
```

```
NEW.doctorID,
```

```
CAST( CONCAT(NEW.date, ' ', NEW.startTime) AS DATETIME ),
```

```
CAST( CONCAT(NEW.date, ' ', NEW.endTime) AS DATETIME )
```

```
) THEN
```

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Appointment clashes with an
existing appointment!';
```

```
END IF;
```

```
END
```

```
$$
```

```
DELIMITER ;
```

2. TRIGGERNAME: Doctors_BEFORE_INSERT

TABLE: Doctors

TIME : BEFORE

EVENT: INSERT

DEFINATION:

DELIMITER \$\$

CREATE TRIGGER `Doctors_BEFORE_INSERT` BEFORE INSERT ON `Doctors` FOR EACH ROW BEGIN

if new.Experience > 5 then

set new.rating = 1;

else if new.Experience <=5 and new.Experience >1 then

set new.rating = 2;

else

set new.rating = 3;

end if;

end if;

END

\$\$

DELIMITER ;

3. TRIGGERNAME: Stock_BEFORE_INSERT

TABLE: Doctors

TIME : BEFORE

EVENT: INSERT

DEFINATION:

DELIMITER \$\$

CREATE TRIGGER `Stock_BEFORE_INSERT` BEFORE INSERT ON `Stock` FOR EACH ROW BEGIN

set new.date = now();

END

\$\$

DELIMITER ;

4.4: Functions:

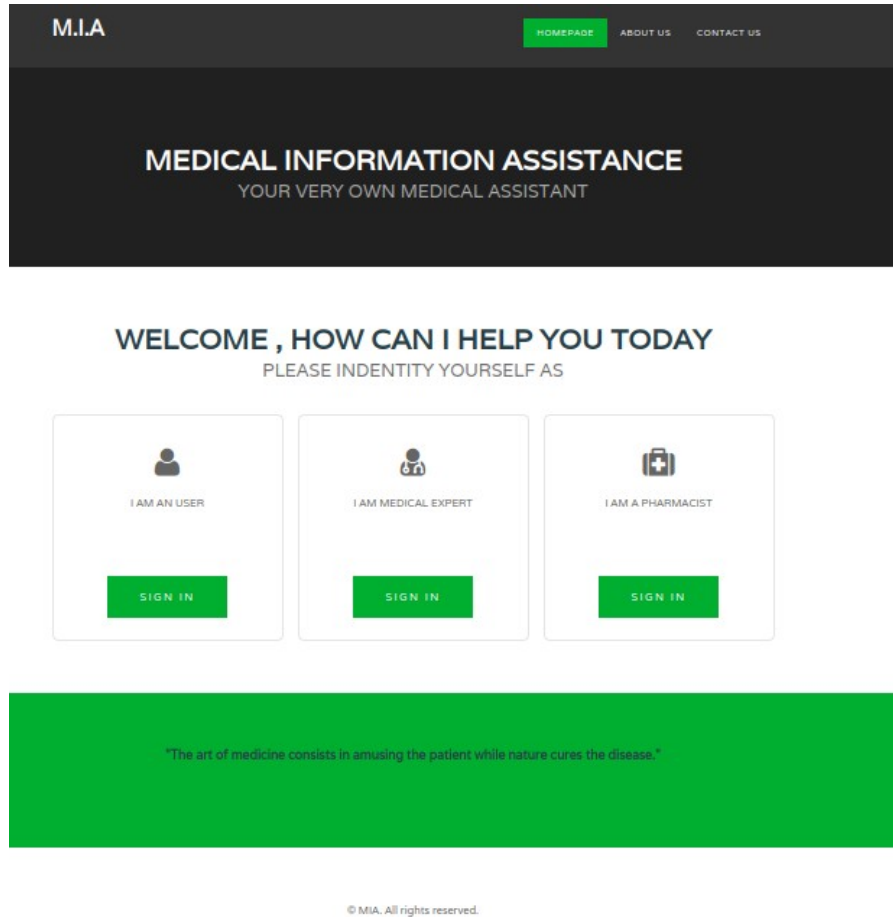
A stored function is a set of SQL statements that perform some operation and return a single value.

Function used in project is as follows

```
CREATE DEFINER=`root`@`localhost` FUNCTION `slotIsAvailable` (`doctorID` INT,
`slotStartDateTime` DATETIME, `slotEndDateTime` DATETIME) RETURNS TINYINT(1) BEGIN
    RETURN CASE WHEN EXISTS (
        -- This table will contain records iff the slot clashes with an existing appointment      SELECT
TRUE
        FROM Appointment AS a
        WHERE
            CONVERT(slotStartDateTime, TIME) < a.endTime -- These two conditions will both hold iff
the slot overlaps      AND CONVERT(slotEndDateTime, TIME) > a.startTime -- with the existing
appointment that it's being compared to      AND a.doctorID = doctorID
            AND a.date = CONVERT(slotStartDateTime, DATE)
        ) THEN FALSE ELSE TRUE
    END;
END$$
```

User Interface

Home Page



Doctor Sign Up Page

MIA

[Home](#) [About Us](#) [Contact Us](#)

Sign Up
Please fill in this form to create an account.

Doctor ID
Enter Doctor ID

Password
Enter Password

Name
Enter Name

Specialization
Enter Specialization

Experience
Experience

Qualification
Enter Education

Clinic
Enter Clinic

Patient ID
Enter Patient ID

Contact
Enter Contact number

Already have an account ? Click Here [Login](#).

Sign Up

Doctor Log In Page

MIA

[Home](#) [About Us](#) [Contact Us](#)

Please fill in this form to sign into your account.

Doctor ID

Enter Doctor ID

Password

Enter Password

New Member ? [Register Now](#).

Sign In

Doctor Home Page

MIA

Home

Profile

Search

Medicine

Appointments

Logout

Home Page

Welcome back, Meenakshi!

Your Current Number of Patients

4

User Sign In Page

MIA

Home

About Us

Contact Us

Sign Up

Please fill in this form to create an account.

Patient ID

Enter Patient ID

Password

Enter Password

Name

Enter Name

Contact

Enter Contact

Insurance ID

Insurance ID

Medical_Information

Enter Medical_Information

Doctor ID

Enter Doctor ID

ALready have an account ? Click here [Log In.](#)

Sign Up

User Log In Page

MIA

Home

About Us

Contact Us

Please fill in this form to log into your account.

Patient ID

Enter Patient ID

Password

Enter Password

New Member ? [Register Now.](#)

Sign In

User Home Page

MIA

Home

Profile

Search

Appointments

Logout

Home Page

Welcome back, Ali!

Pharmacist Sign In Page

MIA

[Home](#) [About Us](#) [Contact Us](#)

Sign Up
Please fill in this form to create an account.

Licence

Password

Name

Address

owner

Already have an account ? Click here [Sign In](#).

Pharmacist Log In Page

MIA

[Home](#) [About Us](#) [Contact Us](#)

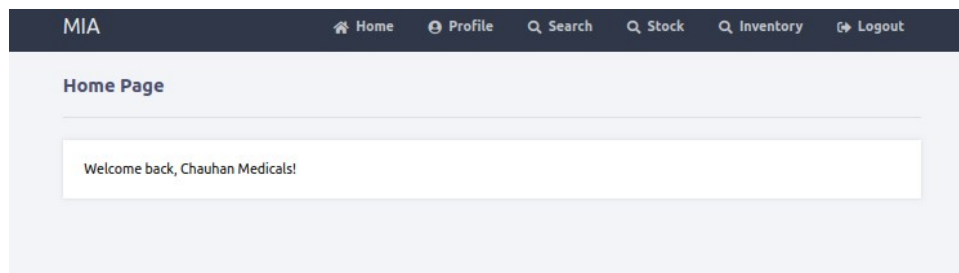
Please fill in this form to sign into your account.

Licence

Password

New Member? [Register Now](#).

Pharmacist Home Page



Conclusion

This system provides an easy way for the operator to interact with the database and to manipulate the data in the database. It has been developed with simple user interactions, efficient and less time consuming. It is helpful to perform paperless work and manage all data. It provides easy, accurate, unambiguous and faster data access.

My project is only a humble way to satisfy the needs of today's medical industry. Several user-friendly coding has also been adopted. I have tried implementing few major features of a medical database. More number of features can be added to this system to make it more efficient.

REFERENCES

1. Python and Python MySQL :

- Introduction to Computer Science and Programming in Python by Dr. Ana Bell, Prof. Eric Grimson and Prof. John Guttag
- Dive Into Python by Mark Pilgrim
- <https://www.python.org/doc> (documentation references)
- <https://www.w3schools.com>

1. Fundamentals of database systems 7th Edition, Ramez Elmasri and Shamkant B. Navathe

2. Web Programming W.B.Sabesta-3rd Edition

3. HTML and CSS : W3.CSS SUCCINCTLY by Joseph D. Booth